

Amendments to the Claims

1. (Currently amended) A program execution method, for transferring, from an interpreter process to a compiled code process, a method that is currently being executed for code that includes a plurality of transfer points at which program execution is transferred from the interpreter process to the compiled code process, comprising the steps of:

moving said transfer points ~~for code~~ to the top of a loop process if they can be moved there without a problem occurring when no problem occurs, even when said transfer points are moved to said top of said loop process;

copying code from the top of the loop process to a point that post-dominates said top of said loop process and said transfer points to a location immediately preceding said loop process, ~~when if~~ said transfer points are located inside said loop process, ~~a point that post-dominates said top of said loop process and said transfer points~~;

storing information for generating recalculation code for specific transfer points when the ~~moving of said code and privatization, and a common sub-expression elimination, and moving of code~~ that are performed pass beyond said specific transfer points; and

performing a recalculation during a transfer process.

2. (Original) The program execution method according to claim 1, further comprising a step of:

defining as a new transfer point, a point from said interpreter process to said compiled code process whereat, when said method that is currently being executed is replaced, the execution speed is increased compared with when said method is not replaced.

3. (Currently amended) The program execution method according to claim 1 or 2, further comprising the steps of:

generating information required to perform a transfer from said interpreter process to said compiled code process; and

storing said ~~obtained~~ generated information while correlating said ~~obtained~~ generated information with said transfer points,

wherein, at said recalculation step, said information stored for said transfer points is employed.

4. (Currently amended) A program for transferring, from an interpreter process to a compiled code process, a method that is currently being executed for code that includes a plurality of transfer points at which program execution is transferred from the interpreter process to the compiled code process, comprising the steps of:

A131
moving said transfer points ~~for code~~ to the top of a loop process if they can be moved there without a problem occurring ~~when no problem occurs, even when said transfer points are moved to said top of said loop process;~~

copying code from the top of the loop process to a point that post-dominates said top of said loop process and said transfer points to a location immediately preceding said loop process, ~~when if~~ said transfer points are located inside said loop process, a point that post-dominates said top of said loop process and said transfer points;

storing information for generating recalculation code for specific transfer points when the ~~moving of said code and privatization, and a common sub-expression elimination, and moving of~~ code that are performed pass beyond said specific transfer points; and

performing a recalculation during a transfer process.